www.icgst.com

A Hybrid Controller for Inflight Stability and Maneuverability of an Unmanned Aerial Vehicle in Indoor Terrains

C. Todd[1], H. Koujan[2], S. Fasciani[3]
[1]*Faculty of Computer Science, Hawaii Pacific University, 1164 Bishop Street, Honolulu, Hawaii, 96813, USA*
[2, 3]*Dept. Information Science and Engineering, University of Wollongong in Dubai, Knowledge Village, Dubai, UAE*
[1]cghourani@hpu.edu, [2]hk963@uowmail.edu.au, [3]stefanofasciani@uowdubai.ac.ae
[1]http://www.hpu.edu, [2, 3] http://www.uowdubai.ac.ae

Abstract

A hybrid controller is designed and tested in simulation-based experiments, overcoming several research challenges associated with stability and control of UAV flight in a GPS-denied environment, achieving faster more accurate autonomous indoor flight. There exists a need for design and modeling of a robust, innovative flight controller that offer more stable, autonomous flight in challenging indoor environments. Drone drifts caused by air turbulence pose as a significant challenge for autonomous aerial systems that are GPS-denied. A hybrid flight controller, that utilizes PID, PIDD^2 and FLC techniques in addition to KF, EKF and Complimentary Error Minimization algorithms, has been developed in a simulation software and validated. Major contributions of the work include: multiple switchable flight modes at the Position and Tracking Controller (PTC) level, and multiple quadcopter flying configuration at the Motor Mixer level. A novel drift correction mechanism utilizes the drone states and an EKF estimator generates a compensation signal with integration of the PTC. FLC adds an extra layer of control by determining the drone's flight mode and flying configuration for the optimal flight output and stability performance. Contributions of this work may further offer increase in accuracy for connected processes enabling UAV flight, including SLAM, path planning, collision detection and avoidance and, subsequently, overall flight performance. [1]

Keywords: *Autonomous UAV, Flight Controller, Drift Correction, Hybrid Controller, Fuzzy Logic and PID Controller.*

1. Introduction

Unmanned Aerial Vehicles (UAVs) are those capable of flight controlled by an onboard computer and/or remotely from the ground. The UAV can be designed to support numerous tasks such as surveillance, scientific research, construction and military operations [1]. UAVs can be classified based on their lift mechanisms such as fixed wings, rotorcrafts or Vertical Take Off and Landing (VTOL) [2]. A quadcopter is a multirotor aircraft built using four thrust sources, generally fixed angle propellers driven by DC motors [2]. Compared to other types of aerial vehicles, quadcopters offer several advantages: the single rigid body with four fixed angle propellers provides a simple mechanical structure, while other aircrafts such as helicopters use one rotor and require complex mechanical components to change the rotor orientation while providing thrust to maneuver [3]. A quadcopter can execute movements with higher accuracy than helicopters, essential when operating in indoor environments [3]. Quadcopters can fly at low speed, whereas fixed-wing aircrafts do not present such flexibility in aerodynamics and require high air pressure for flight capability that occurs only at a high speed [3]. However, quadcopters present major drawbacks in terms of power efficiency, due to continuous, rapid acceleration of the four motors and flight stability. Effective control strategies may be introduced for stability improvements [4].

In this work, a novel approach to controller design is proposed toward enhanced UAV (Quadcopter) flight stability and maneuverability, overcoming several research challenges within this field. A Quadcopter possesses structural characteristics that change with its dynamic model and each model poses unique design challenges. Appropriate modeling of aerodynamics and dexterous control are the main concerns in this field. To establish autonomous flight accuracy of pose estimation, obstacle location and classification are key issues [5][6]. Indoor environments raise special concerns in terms of nature of failures and safety issues. Current UAV flight controller techniques can handle sparse and large indoor areas, but embedded and/or moving objects of various type and shape still place significant restrictions on indoor drone flight capability. This context requires sophisticated

[1] This study has been implemented on a Simulink platform in Matlab.

and advanced control algorithms to circumnavigate obstacles, for path planning and localization. Sophisticated control algorithms may be designed to enable a UAV to circumnavigate indoor embedded obstacles, providing dexterity in control strategy. Indoor spaces deny the UAV's accessibility to signals provided by the Global Positioning System (GPS), and as such, alternative solutions, for drone localization and/or as input for control, are needed. A control system that communicates with path planning and collision avoidance is required to output correct flight path decisions for the drone's controller and also to facilitate UAV inflight stability, which is subject several disturbances in indoor contexts [1], [4], [6], [7], [8]. Air turbulence reflected from objects close to the drone, including when taking off or landing, causes lateral drifts that affect the accuracy of the autonomous functions [1], [7]. Light-weight UAVs introduce additional challenges in dynamic modeling because the assumption made for larger drones, such as structural rigidity and symmetry, are no longer valid and non-linear parameters in the mathematical model of the Quadcopter frame should be introduced [6]. Simultaneous changes in the drone's motors speeds add another level of complexity that may determine in instability [8].

For UAV autonomous control, research challenges are of a technical nature and are application-specific, such as relating to drift, instability and inaccuracies due to inappropriate fine-tuning of the flight controller. This work overcomes associated research challenges by designing a hybrid flight controller system offering stability and high accuracy in indoor autonomous flight. A control algorithm that achieves better flight stability for indoor autonomous quadrotors as compared to existing solutions [9], [10], [11], [12] is designed, implemented and validated using a hybrid model of Proportional, Integral, Derivative (PID) and Fuzzy Logic Control (FLC), in Matlab Simulink. UAV design exhibits improved performance offering faster response time and accuracy of flight control, in comparison to existing techniques [1], [2]. This system overcomes disturbances of drift in which better results of mapping, localization, collision detection and avoidance may be obtained, within GPS-denied environments.

Section (2) focuses on the hybrid controller design and implementation in Simulink, in Matlab. Section (3) presents the results of the study. Section (4) provides a discussion of the research findings and Section (5) emphasizes major conclusions and recommendations of the work.

2. Controller Design and Implementation

In the applied method, it is assumed that the drone has full operability of all propellers and that the battery provides supply power for the duration of the flight. Environmental issues such as lateral drifts during drone-obstacle air turbulence interference are considered for controller design. A standard quadcopter is considered, as in [1], [7], [13], with rigid body construction and four thrust sources (motors) placed on each wing, equi-spaced from the frame center, as in Figure 1. Propeller rotation direction plays an important role in achieving quadrotor basic stability [7]; each two opposing fans, separated by 180^o, should rotate in the same direction, with direction opposite to that of the other fan pair. For example, if propellers 1 and 3 are rotating clockwise, propellers 2 and 4 should rotate anti-clockwise [13]. Quadrotor motion is described with its tilt on three axes of rotation w_x ,w_y and w_z [13]. The UAV rotation around the w_z (Yaw) axis results in turning to the left or right, and it can be achieved by reducing the speed of one of the counter rotating propeller pairs [1]. Rotation of the quadrotor about the w_y (Pitch) axis is performed by rising the angular speed of motor 4 and decreasing the speed of motor 2 [1]. Rotation about the w_x (Roll) axis is controlled by the same operation as for Pitch but with fans 1 and 3 [1]. The UAV's take-off and landing mechanism simultaneously increases or decreases the speed of all propellers [2]. Basic maneuvers of a quadrotor assist in understanding stability requirements and flight control of a UAV of this type. The controller design proposed controls six degrees of freedom including yaw, pitch, roll, x-transition, y-transition and attitude of the four motors positioned on the quadrotor. This is achieved through three layers of control.

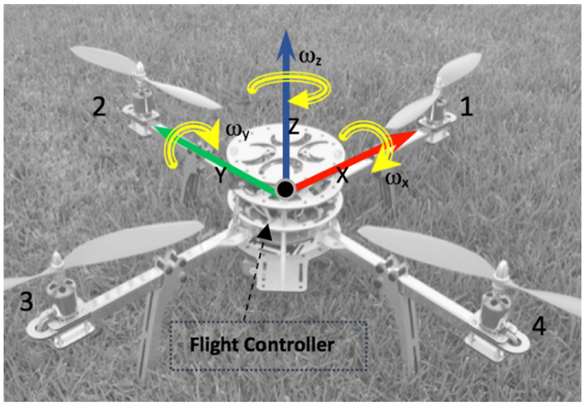

Figure 1. Quadcopter: six Degree of Freedom, rigid frame with four equi-spaced thrust sources.

PID control is combined with FLC to provide a hybrid solution for greater quadrotor control and stability during indoor autonomous flight. Several PID controllers are embedded enabling multiple inputs for more complex functions demanded by the drone [9]; a technique that offers a higher control accuracy of each degree of freedom of the quadcopter [8], [10]. The FLC requires that control functions are written using linguistics for all possible control scenarios, forming the control data sets [5]. Compared to other methods, FLC do not require an accurate input signal: smaller errors and overshoots of system output, and faster time to reach steady state produce better results in terms of final stability [2], [14]. The Backstepping control technique is not considered in this design due to the high computational complexity, which is not suitable for UAVs controlled by onboard low power computational platforms. Backstepping is a recursive complementary tool that has been used with other control techniques to achieve better stability than singular control techniques [5], [10], [15]. A major challenge in designing and implementing the Backstepping controller is the greater complexity and accuracy in terms of the mathematical and dynamic models required to fully integrate them in a quadrotor. The

Backstepping controller is substituted by the Kalman Filter (KF) and the Extended Kalman Filter (EKF) that enable estimating the result of a process with a high degree of confidence. The EKF algorithm is designed to estimate the state of a non-linear systems affected by unpredictable noises. EKF uses less computational resources and it requires less information about the estimated system than a Backstepper controller [16]. The aforementioned techniques have been implemented to minimize the measurement errors and to estimate the drift of the quadcopter as part of the drift correction mechanism.

A hybrid controller that exploits the advantages of all the three controllers: PID, FLC and EKF has been designed, developed and tested in this work. PID controllers are designed to control independently pitch, roll, yaw and altitude; one controller is dedicated to each degree of freedom. The FLC is designed to then achieve further flight stability; it continuously adjusts the flight mode and the motor control mixer configuration. Further, the FLC operates as the main controller that determines the movement output of the UAV based on two parameters; the input signal and the estimated disturbance existing in the drone's environment. An overview of the proposed system is illustrated in Figure 2.

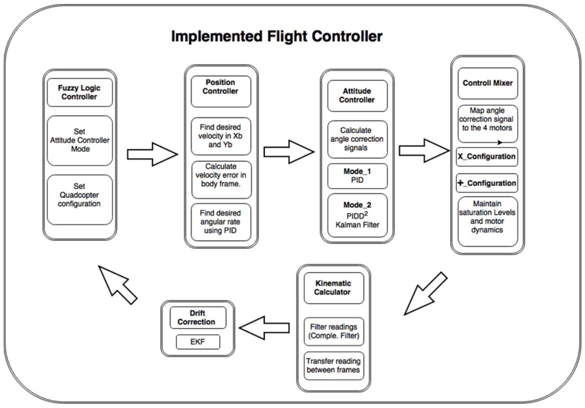
Implemented Flight Controller
Fuzzy Logic Controller
Set Attitude Controller Mode
Set Quadcopter configuration
Position Controller
Find desired velocity in Xb and Yb
Calculate velocity error in body frame.
Find desired angular rate using PID
Attitude Controller
Calculate angle correction signals
Mode_1 PID
Mode_2 PIDD^2 Kalman Filter
Controll Mixer
Map angle correction signal to the 4 motors
X_Configuration
+_Configuration
Maintain saturation Levels and motor dynamics
Kinematic Calculator
Filter readings (Comple. Filter)
Transfer reading between frames
Drift Correction
EKF

Figure 2. Configuration of the FLC, PID and EKF controllers within the hybrid controller design.

The drift correction algorithm (Figure 3) designed for this flight controller, utilizes an EKF to estimate the real pitch and roll of the drone away from the control processes. Then, the estimated angles are fed back to the flight controller as an error signal after subtracting them from the input command signals. This error, the difference between the two signals, represents the drift of the quadcopter that will be corrected automatically by the flight controller as it computes the other needed corrections.

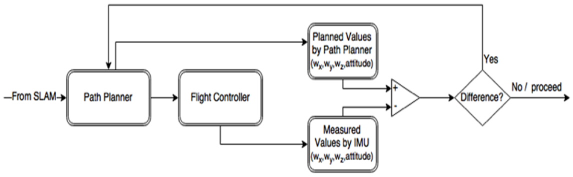
From SLAM
Path Planner
Flight Controller
Planned Values by Path Planner (w_x,w_y,w_z,attitude)
Measured Values by IMU (w_x,w_y,w_z,attitude)
+
-
Difference?
Yes
No / proceed

Figure 3. Flowchart of the algorithm for Drift Correction.

Modeling of Drone Control: The controller utilizes multiple techniques in control and filtering including PID, Proportional, Integral, Derivative (Second Order) PIDD^2, FLC, KF, complementary filtering and Extended Kalman Filter (EKF). The flight controller is divided into five components: Position and Tracking Controllers, a Motor Mixer, a Drift Correction mechanism and the FLC. Novel techniques are included in each component of the UAV flight control and stabilization: the Position Controller compensates for drift correction by application of the EKF in addition to applying existing techniques including basic manipulation of the desired x-, y- and z- coordinate, which represent the desired rotational rate as output. The Tracking Controller receives the output from the Position Controller and together with the proposed control strategies, determines the correction signal required and adjusts the quadcopter's attitude. Existing techniques implement only a single control methodology for the Tracking Controller for all flight conditions. In this work the Tracking Controller has two flight modes with dynamic switching capability for mode selection, to overcome induced stability and fast-response issues. The motor control mixer then maps the angle correction signals determined by the Tracking Controller into the required throttle for each of the four motors on the UAV frame. The quadcopter can fly in two different kinematic configurations: X configuration, in which the drone's X-Y axes of movements are 45 degrees apart between each of the four motors and the Plus (+) configuration, in which the drone's body frame is the same as its inertial one. In this work, there is an option to switch between configurations based on the desired output which is a novel approach.

For autonomous drone flight, drifts are key concerns. To address these, the EKF has been designed and implemented to estimate environmental drifts. This drift correction mechanism estimates pitch and roll angles in isolation from control process noise and interference; these estimates are then compared to the planned angles for drift calculation. The compensation commands are then generated by the Position Controller. PID and associated derivatives are implemented in the Flight Controller for individual motor control. To improve system stability, the FLC is implemented to determine the optimal setting required to drive the most stable output: given the multiple operating flight modes and configurations of the Flight Controller, the FLC operates to drive mode and configuration selection for greatest stability.

Mechanical Modeling: Mathematical modeling of quadcopter aerodynamics enables calculation of the required practical parameters for controller implementation. Two conventional coordinate systems are for the quadcopters are the Plus (+) configuration, with the x- and y- axes as the arms supporting the two consecutive motors (90° arm separation). According to the Plus convention the first motor must rotate anti-clockwise and the second motor clockwise. In the Ex (X) configuration, the x-axis is aligned by +45° from the arm of the first motor towards the second; that is, the Plus configuration rotated by +45°. Changes in configuration will cause differences in modelling and alignment of motors during this process. In this work, the both configurations are enabled with switching between each mode.

Assuming rigidity of the quadcopter's structure and propellers, that the UAV structure is symmetrical, and the center of gravity and fixed body frame coincide, the moment of inertia of the quadcopter mass may be defined by the following matrix (Equation 1):

$$J^b = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}$$

Equation 1

By applying approximations [11], [12], the momentum of the quadcopter's structure, actuator mass and arm mass are determined by Equations 2, 3 and 4.

$$J_{xx} = J_{x,Motor} + J_{x,Hub\,Body} + J_{x,Arm}$$

Equation 2

$$= \Bigg[2\left[\frac{1}{4}m_M {r_M}^2 + \frac{1}{3}m_M {h_M}^2\right]$$

$$+ 2\left[\frac{1}{4}m_M {r_M}^2 + \frac{1}{3}m_M {h_M}^2\right.$$

$$\left.+ m_M {d_M}^2\right]\Bigg]$$

$$+ \left[\left[\frac{1}{4}m_H r^2 + \frac{1}{12}m_H {h_H}^2\right]\right]$$

$$+ \Bigg[2\left[\frac{1}{2}m_A {r_A}^2\right]$$

$$+ 2\left[\frac{1}{4}m_A r^2 + \frac{1}{3}m_A {L_A}^2\right.$$

$$\left.+ m_A {d_A}^2\right]\Bigg]$$

$$J_{yy} = J_{y,Motor} + J_{y,Hub\,Body} + J_{y,Arm}$$

Equation 3

$$= \Bigg[2\left[\frac{1}{4}m_M {r_M}^2 + \frac{1}{3}m_M {h_M}^2\right]$$

$$+ 2\left[\frac{1}{4}m_M {r_M}^2 + \frac{1}{3}m_M {h_M}^2\right.$$

$$\left.+ m_M {d_m}^2\right]\Bigg]$$

$$+ \left[\left[\frac{1}{4}m_H r^2 + \frac{1}{12}m_H {H_H}^2\right]\right]$$

$$+ \Bigg[2\left[\frac{1}{2}m_A {r_A}^2\right]$$

$$+ 2\left[\frac{1}{4}m_A r^2 + \frac{1}{3}m_A {L_A}^2\right.$$

$$\left.+ m_A {d_A}^2\right]\Bigg]$$

$$J_{zz} = J_{z,Motor} + J_{z,Hub\,Body} + J_{z,Arm}$$

Equation 4

$$= 4\left[\left[\tfrac{1}{2}m_M {r_M}^2 + m_M {d_m}^2\right]\right] + \left[\tfrac{1}{2}m_H {r_H}^2\right] + \left[4\left[\tfrac{1}{4}m_A {r_A}^2 + \tfrac{1}{3}m_A {L_A}^2 + m_A {d_A}^2\right]\right]$$

where m_x is the mass of the sub-notation *x* which could be referring to the motor (*M*), drone's arm (*A*) or drone's hub body (*H*), d_x is the radius from what is referred to in the sub-notation *x* to the Centre of Mass, h_x is the height of sub-notation x from its base, L_x is the total length from the base to sub-notation x, and r_x is the radius of the structure referred to in the sub-notation x.

The main forces and moments that affect the drone are from gravity or due to the thrust and torque initiated by the four rotors [3]. The thrust generated by the propellers are conventionally forces that are perpendicular to the x-y plane of the body frame in the positive direction of the z axes [12]. Drone maneuverability is constrained by motor thrust, which is the key driving force for its movements [11]. The generated thrust, *T*, by an actuator on the quadcopter is described by Equation 5.

$$T = C_T \rho A_r r^2 \varpi^2$$

Equation 5

where C_T is the thrust coefficient provided by the manufacturer of the rotor or measured manually, ρ is the air density, A_r is the propeller's rotation cross-sectional area, r is the radius of the rotor and ϖ is its angular velocity. Simplifying Equation 5, Equation 6 represents the lumped thrust coefficient, c_T.

$$T = c_T \varpi^2$$

Equation 6

The total thrust provided by all four propellers is represented by Equation 7. Varying the thrusts of motors 2 and 4 results in rotation of the drone about the x-axis. This rotation is coined the rolling torque, *L*, as represented by Equation 8.

$$T_{tot} = T_{motor1} + T_{motor2} + T_{motor3} + T_{motor4}$$

Equation 7

$$L = l(T_{motor4} - T_{motor2})$$

Equation 8

where l is the arm length of the drone. The pitch torque, *M*, is the drone's rotation around y-axis and results due to a change in the thrust of motors 1 and 3. Pitch torque is calculated by application of Equation 9.

$$M = l(T_{motor1} - T_{motor3})$$

Equation 9

The yawing torque, *N*, is the drone's rotation around the z-axis. It is the resultant of the difference in the developed torques between all four motor rotations (counter-clockwise and clockwise) and is expressed as Equation 10.

$$N = l[(T_{motor2} + T_{motor4}) - (T_{motor1} + T_{motor3})]$$

Equation 10

The torque generated by each motor, *Q*, can be expressed using the lumped torque coefficient, and is represented by Equation 11.

$$Q_{motor\_x} = c_Q {\varpi^2}_{motor\_x}$$

Equation 11

Practical tests enable determination of the parameters in Equations 1 to 11. All inertial forces and torques of the quadcopter may then be grouped into one, inertial, matrix (Equation 12).

$$\begin{bmatrix} \Sigma T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & d_+ c_T & 0 & -d_+ c_T \\ -d_+ c_T & 0 & d_+ c_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \\ \varpi_3^2 \\ \varpi_4^2 \end{bmatrix}$$

Equation 12

where *d* is just another convention notation for the aforementioned notation *l*.

For modeling of the mechanical system, all parameters and forces that affect the aerodynamics of the UAV may be captured in one matrix. The resulting matrix, $M_{A,T}^b$, (Equation 13) accounts for all of the aerodynamic, gyroscopic, and thrust moments created by the rotors' systems on the quadcopter for the Plus configuration. The lift force is represented by Equation 14. Additional effects such as blade flapping and frame aerodynamic drag not included in the model.

$$M_{A,T}^b = \begin{bmatrix} d_+ c_T \varpi_2^2 - d_+ c_T \varpi_4^2 + J_m Q \left(\frac{\pi}{30}\right) (\varpi_1 - \varpi_2 + \varpi_3 - \varpi_4) \\ -d_+ c_T \varpi_1^2 + d_+ c_T \varpi_3^2 + J_m P \left(\frac{\pi}{30}\right) (-\varpi_1 + \varpi_2 - \varpi_3 + \varpi_4) \\ -c_Q \varpi_1^2 + c_Q \varpi_2^2 - c_Q \varpi_3^2 + c_Q \varpi_4^2 \end{bmatrix}$$

Equation 13

PID Controller Model: The quadcopter PID flight controller consists of two sub-systems: the first determines the desired quadcopter rotational rate driving position control, and the second measures the actual quadcopter rotational speed for calculation of the required motor output to achieve the desired rotational movement, as shown in Figure 4. The Inertial Measurement Unit (IMU) attains actual rotational position information pertaining to the rotational speed of the drone in addition to the angle of deviation from any of the drone axes. This actual angle measurement is compared with the desired angle of the drone, which represent the input of the first PID controller. The difference of desired rotational rate to meet the requirements is sent to the second PID controller and compared with the actual rotational rate to provide motors with a throttle control signal (voltage).

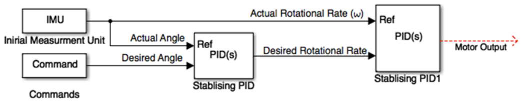

Figure 4. Diagram of the two sub-systems for PID control of motor angular rotation.

Dynamic alternation is achieved in-flight between different stability and structural modes. The Tracking Controller processes the quadcopter commanded desired position in two stages: first it finds the feed forward error of the quadcopter's position to determine the desired rotational velocities, and then it finds the required correction to be performed by the drone for its rotational states and altitude. The Motor Mixer Controller translates the required corrections into throttle commands that are distributed to each of the four motors and are dependent on the quadcopter configuration. Both Plus and Ex configurations integrated in the system, requiring a switching strategy to provide maximum drone maneuverability. The State Finder manipulates all parameters, state equations and forces acting on the drone for current use by the system.

The Tracking Controller determines the correction needed for pitch, roll and yaw of the drone, in addition to the altitude, for velocity control; by comparing actual and desired position and attitude readings throughout several layers of PID control. An improved controller is achieved over existing solutions [9], [10], [11], [12] with novelty in utilization of two modes (configurations) in combination with FLC for tuning and a PIDD^2 controller with a KF to filter out noise. This approach further provides a more accurate measurement of the angle of the quadcopter. Variation of the Tracking Controller from existing designs lie in the dual mode switching capabilities dependent on the flight environment, in addition to the KF implementation to calculate more exactly the error signal of the feed-forward PIDD^2. The KF effectively eliminates noise due to measurement and control to increase the accuracy of the correction signal made by the controller.

After receiving the desired x and y coordinates, altitude and yaw, the Tracking Controller calculates the desired pitch and roll angles from these inputs, applying Euler's Equations to compute the transformation of the drone's position and velocities between its inertial- and body-frames. First the error is determined for the x and y coordinates of the body-frame. This error is then mapped to a desired velocity in the same frame, which is transferred to the desired rotational rate. For calculation of error and its mapping to desired rotational rates, the yaw (Psi angle, Ψ) is needed since a quadcopter can maneuver towards a point with different Yaw angles. The x and y components of the yaw are scaled by the difference between the current coordinates and intended ones to find the desired velocity in the x and y axes of the body frame, as in Equation 15 and 16 [3]. Figure 5 shows the Simulink design of the Tracking Controller.

$$U_{Xb,Desired} = X_{error} * \cos(\psi) + Y_{error} * \sin(\psi) = (X_{cmd} - X_{actual}) * \cos(\psi) + (Y_{cmd} - Y_{actual}) * \sin(\psi) \qquad \text{Equation 15}$$

$$\mathrm{U}_{Yb,Desired} = Y_{error} * \cos(\psi) - X_{error} * \sin(\psi) = (Y_{cmd} - Y_{actual}) * \cos(\psi) - (X_{cmd} - X_{actual}) * \sin(\psi) \qquad \text{Equation 16}$$

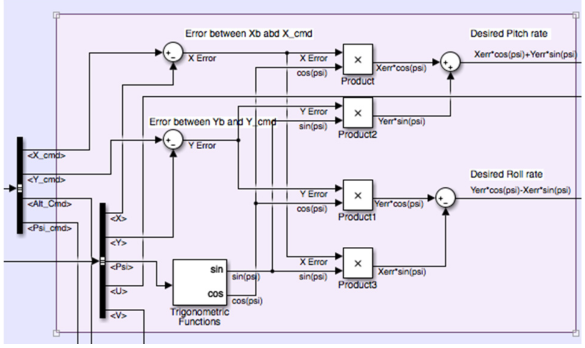

Figure 5. Simulink design for modeling of the mathematical manipulation needed to determine the desired velocity correction in x and y axes of the body-frame using the signals coming from the commands and quadcopter state manipulator.

Actual pitch and roll rates are controlled by PID controllers that use the desired velocities and actual velocities, *U* and *V*, measured by the IMU. The feed-forward PID technique is applied since it offers faster response than a feedback PID system for control of electric motors [15], [17]. The feed-forward subsystem calculates the required adjustment needed by each rotational drone axis; this is sent as a command signal to the next subsystem to direct drone movement by the quantified angle about the respective axis. For calculation of the angle of required rotation *Theta*, the difference between the pitch rate and the linear velocity is calculated along the x-axis of the inertial-frame. This provides the velocity error, which is sent to a feed-forward PD controller to compute the required *Theta*, limited within the physical characteristics of the drone and input to the next section of the controller. For calculation of the angle of rotation about the y-axis, the roll rate, the desired roll rate comprises the positive input while the negative input is the linear velocity in the y-axis of the inertial-frame. A PD controller enables the required compensation in terms of roll to be found. The sign inversion corrects the calculated error due to mathematical operations.

The pitch θ, roll φ, yaw Ψand altitude correction signals are computed in the following subsystem of the controller using feed-forward PID controllers. This process can be done in one of two modes: the Acrobatic mode for fast, sharp maneuvers or the Extra Stable mode for additional stability. In Acrobatic mode, first the error is determined and is then fed into the proportional and integral components of the controller (Figure 6). This error is the difference between desired and actual the drone's rotational angle measurements. The derivative component comprises a scaling factor since rotational rates of the drone may be determined through analysis of the IMU readings and application of Euler's Equations; substituted as the derivative of the corresponding angle. All four required correction signals are generated via identical controllers with the exception of altitude because this has a gravity offset input to achieve hovering at a specific height. Further, the altitude controller finds the derivative of the linear velocity on the z-axis to calculate the vertical acceleration of the drone which assists in its stabilization at the defined gravity offset.

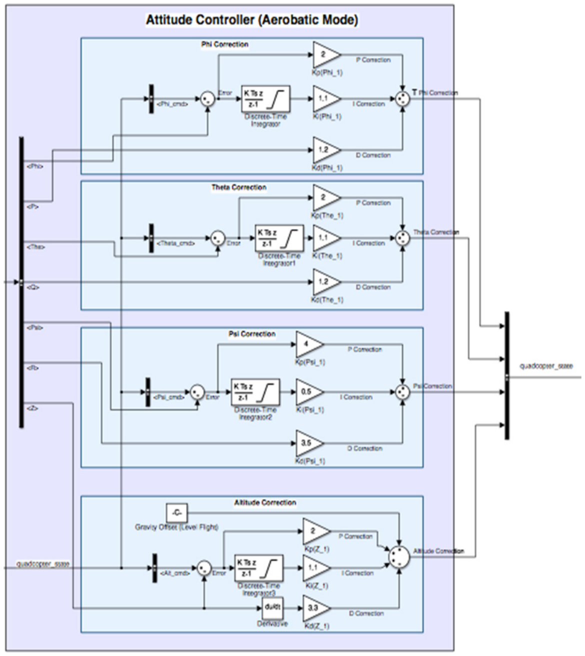

Figure 6. Simulink implementation of the Aerobatic flight controller mode.

The Extra Stable mode for flight control detailed in Figure 7 enables higher flight stability with a tradeoff in slower response time due to the KF applied to the input signal for noise reduction, providing accurate estimates of pitch (θ) and roll (φ) angles. The derivative component in the PID controller projects the future error of the plant signal and also enables the controlled system to stabilize about a targeted point. This type of controller is a PIDD^2 due to the second derivative component and prove useful in flight systems such as this in achieving with greater accuracy and stability, the desired attitude or position. Angle correction calculations in this mode are performed using a similar PIDD^2 structure, with altitude correction requiring an additional component for consideration of initial conditions, such as if the drone had a specified starting altitude (gravity offset).

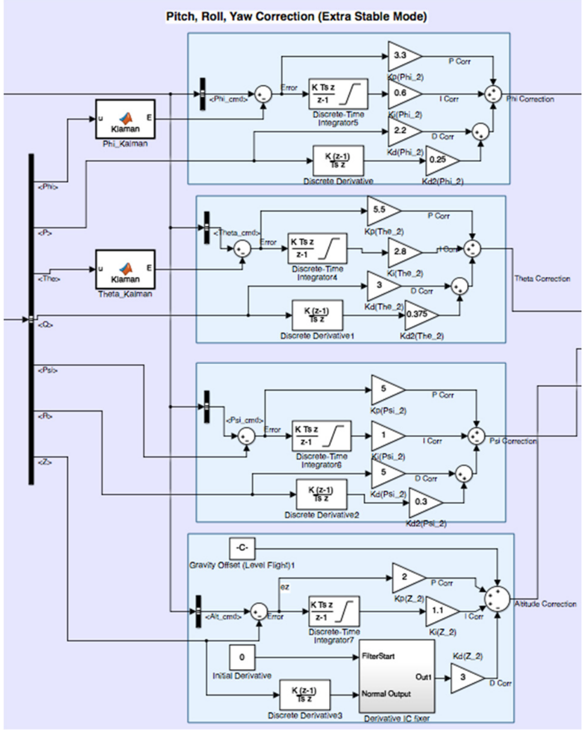

Figure 7. Simulink implementation of the Extra Stable flight controller mode.

The KF is implemented as a Matlab function box, which is a customized Simulink block. The state model of this KF, *A*, includes the angle and it first two derivatives, and is initialized to zero; *A* = [1 dt -dt; 0 1 0; 0 0 1]. This assumes the filter has a coordinated start with the flight controller when the drone is at rest and assumes a balanced start position. The prediction of the error covariance matrix, *P*, is initialized as: *P* = [1000 0 0; 0 1000 0; 0 0 1000]. The KF recursively updates the values of *P* to within reasonable limits from the initially high values. The scaling matrix, *B*, of the control noise is assumed to be zero for the purposes of simulation. P is then updated with consideration of values of *A*, according to the following Matlab script: *P* = *A*P*A*.' Filter observations, *z*, are then equated to the block input signal, *u*, which is the angle measured by the drone. The matrix, *C*, that maps the current drone state to the measurements read by the IMU, which in this case is set to: C = [1; 0; 0], since the filter estimates only the angle and other derivatives are not needed. Calculation of the prior state estimate is then performed using Equation 16, with *x* initialized to 0; *x* = [0; 0; 0]. The filter gain is implemented in a two-step process, by applying Equation 17 to determine *S,* the denominator of the gain equation, and then Equation 18 to calculate gain *K*. *R* is a matrix that contains the sensitivity values of the sensors used to implement the KF, however for simulation *R* is initialized to zero since the simulated IMU signal has no sensitivity; *R* = [0 0; 0 0]. Finally, the filter finds the posterior state estimates (*x*) and updates the error covariance matrix (*P*) using Equations 19 and 20 respectively. The output of this block is the estimated angle by the KF based on the inputted measured angle from the flight controller.

$$y = z - C \text{ x } x$$

Equation 16

$$S = C \text{ x } P \text{ x } C' + R$$

Equation 17

$$K = P \text{ x } C' / S$$

Equation 18

$$x = x + (K \text{ x } y)$$

Equation 19

$$P = (I - (K \text{ x } C)) \text{ x } P$$

Equation 20

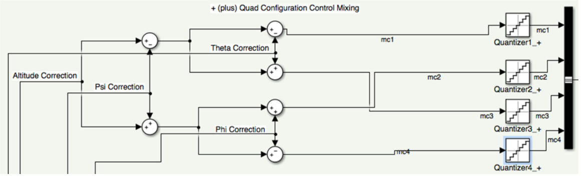

Figure 8. Simulink model of the control mixing signal processing for Quadcopter Plus Configuration.

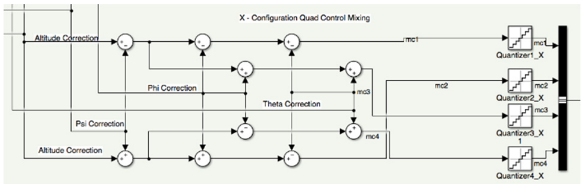

Figure 9. Simulink model of the control mixing signal processing in Ex Configuration.

A control signals mixer has been implemented to provide functions that scale and map the input signal to one that is a meaningful command to the actuator, such as correction signals are translated into throttle commands per motor to drive the quadcopter to the required position. To provide the mixing equations for each of the two configurations

Plus and Ex: changes in speed and required rotational direction of the four propellers map to changes in elevation, pitch roll and yaw. This control mixer varies from existing solutions due to the ability to utilize both Plus and Ex configurations. Further, this model enables dynamic in-flight configuration toggling which widens design capabilities for drone path planning systems and provides a greater range of possible maneuvers, particularly suitable for indoor environments. Motor control signal mixers use the angles correction signals, which have been calculated by the tracking controller as shown in Figures 8 and 9. Quantizer blocks are used to quantize the mixer control signals since real-time processing is a discrete process.

Drift Correction: Lateral drifts have adverse effects on quadcopter indoor SLAM, path planning and control techniques due to drift calculation complexity. Drifts experienced indoors are primarily due to reflected air turbulences from obstructions surrounding the drone such as furniture. EKF algorithms provide estimation of the state of non-linear systems suffering from highly unpredictable noises [16]. In this system, EKF is applied for the drift correction algorithm, illustrated in Figure 10, to estimate the actual pitch and roll of the drone as abstract from the control process. This is also used as feedback signal for the flight controller angle estimator. These are subtracted from the input command signals and their difference estimates the error representing quadcopter drift, which is corrected automatically by the flight controller. The gyroscope and accelerometer have been simulated for calculation of drone pitch and roll angle without interference of noises from the control processes.

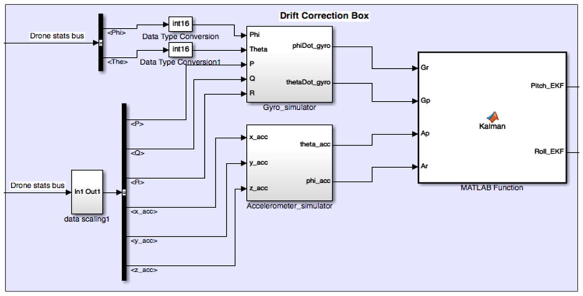

Figure 10. Drift Correction implementation on Simulink: pitch and roll measurement readings of the drone are input to the EKF filter which then estimates the actual angles. The estimated angles are fed back to the flight controller where they are subtracted from the desired angle to give a new angle correction signal that compensates for the non-linear drifts.

Fuzzy Logic Controller: The flight controller has a configuration and several modes that provide improved stability performance over existing systems [2], [14], based on user preferences, yet as flight conditions change and command inputs vary, controller stability performance can degrade. To compensate for these stability variances experienced during changes in flight environment and issued commands, a FLC has been designed and implemented to provide overall improved controller stability performance. During high disturbance greater demands on drift correction are required and the flight controller achieves better flight stability using the Extra Stable mode. When requiring fast and sharp movements the Aerobatic mode outperforms its counterpart. Due to such demands, commanded pitch and roll angles provide function in determining the flight tracking controller mode selected. Further, Ex configuration is able to sustain greater forces due to disturbance than the Plus configuration and as such the former configuration is used to attain greater flight stability in high disturbance conditions. A Mamdani model for the FLC is implemented in Simulink (Figure 11), with commanded pitch and roll angles as input to determine the flight tracking controller mode. In addition, the disturbance is input in the form of values within the covariance matrices of the drift correction EKF to determine the required quadcopter configuration. The four membership functions of pitch and roll inputs are the same; this function is shown graphically (Figure 12 and Figure 13). Membership is assigned based on the current pitch or roll angle that varies from -90 to 90 degrees; if the angle is between -90 and -45 or 45 and 90 it is characterized as high, otherwise it is awarded low. There are two membership functions for disturbance, as shown graphically (Figure 14); if the disturbance is less than 20% of the maximum EKF noise covariance values it is labelled low, while if the disturbance is greater than this value (20%) it is categorized as high.

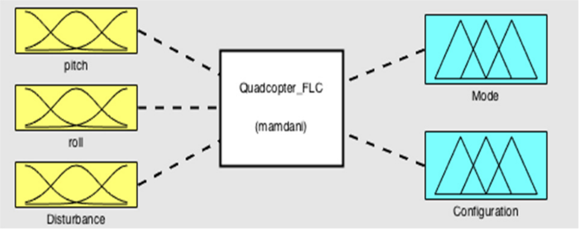

Figure 11. Overall design of the FLC of the flight controller.

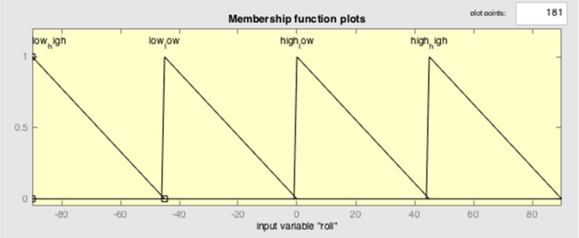

Figure 12. Membership Functions of the Roll input.

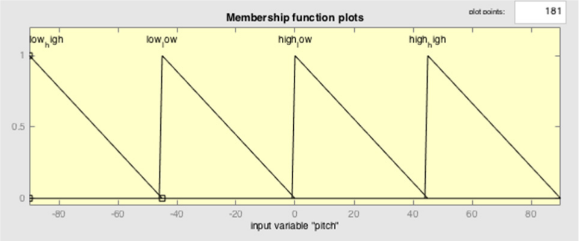

Figure 13. Membership Functions of Pitch input.

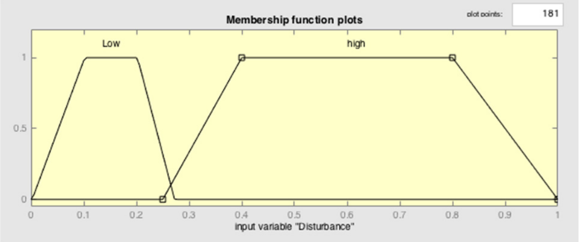

Figure 14. Membership Functions of the Disturbance input.

3. Results

System design and implementation for quadcopter control has been implemented and tested in Simulink and Matlab. System position control is achieved using the Tracking Controller and Mixing Controller. Noises attributed to measurement and control processes are minimized by the KF and Complementary Filter. Flight drifts are calculated by the EKF which are then compensated by the Tracking Controller. The FLC establishes the correct flight mode and drone configuration for optimal stability and dexterity performance amid varying environmental conditions (disturbances, input commands). Results were collected and analyzed from the Position Controller, Tracking Controller with KF, Complementary Filter of the IMU Output, Drift Correction using EKF and FLC Controllers. Performance is evaluated in terms of accuracy, flight stability and error minimization.

Position and Tracking Controller: Controllers were subject to testing during drone flight path change to determine the behavior of the first part of the position controller, in which the desired position is mapped into drone's body frame velocity to calculate the error in the body frame. Figure 15 shows the commanded x-coordinate (in blue) and current x-coordinate measurement (in red), in addition to the output of the position controller (in yellow) for velocity of the body-frame along the x-axis. Testing carried out on the second stage of the controller where error in the body-frame velocities is translated into a desired angle. Figure 16 shows the commanded Theta angle represented (in red) and measured Theta angle (in blue) captured from the test. Test results for the tracking controller in Aerobatic Mode are shown in Figure 17 with the commanded Theta (pitch) angle (in blue), the correction signal (in yellow), and the measured pitch (in red). Figure 18 shows test results for the tracking controller in Extra Stable Mode with the commanded Theta angle (in blue), measured Theta angle (in red) and correction signal (in yellow).

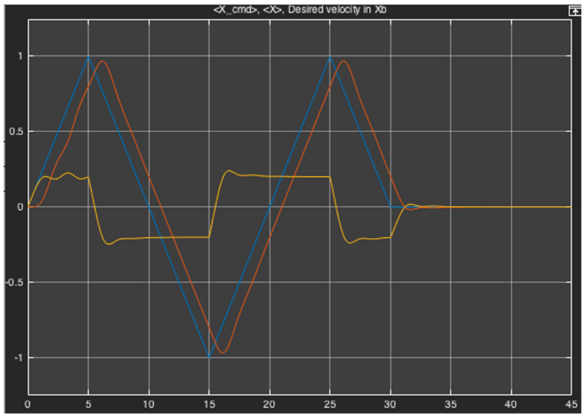

Figure 15. Commanded signal (blue), current measurement (red) and output of the Position Controller for body-frame velocity (yellow); Time versus Position (Radians).

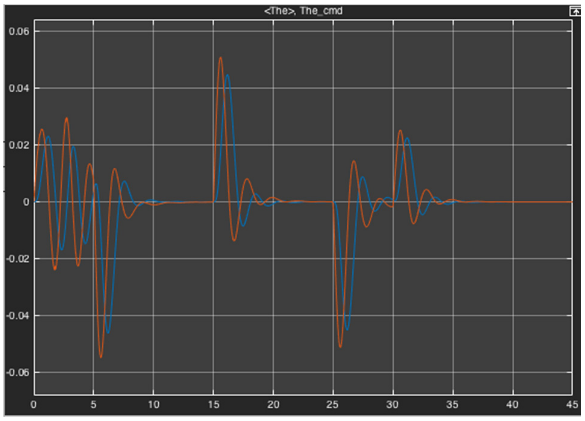

Figure 16. Output of the second stage of the position controller: commanded (red) and measured (blue) signal; Time versus Position (Radians).

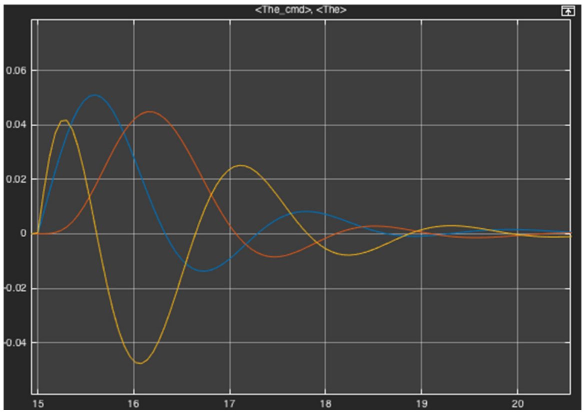

Figure 17. The Theta correction signal of the Aerobatic mode (yellow), the commanded Theta signal (blue) and measured Theta (red); Time versus Position (Radians).

Results captured in Figures 15, 16, 17 and 18 show stable position and tracking controller performance. Drone angles followed the commanded values driven by the correction signals. These were successfully mapped into the proper form of input at the different levels of the controller. Results in Figure 17 reveal that the Aerobatic mode of the tracking controller has a stable bounded output and the drone is following the fast changing theta commands, however, it has a small time lag that could be reduced with further tuning of the PID parameters. Results in Figure 18 related to the Extra Stable mode, show that the controller successfully rejects the control noises present in the time range [2; 4.5]. It is possible to observe that in the same time range the Theta angle is not affected by noise. The noise rejection is a result of using the KF inside this mode. Although the controller caused the drone to exceed the command values when the time is equal to 0.8, it manages to correct this minimizing the difference with the commanded signal. This issue is a result of the KF error covariance matrix which provides less accurate state estimates at the first iterations of the KF. Analysis of the controllers reveals stable and accurate performance, meeting design objectives.

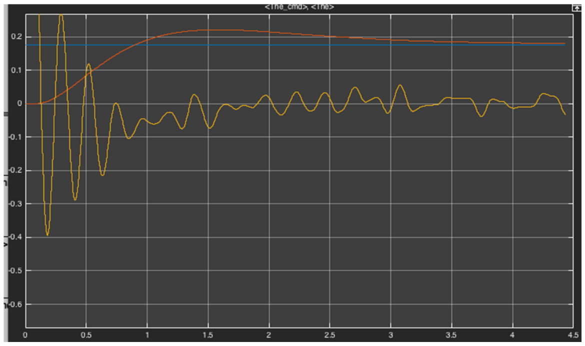

Figure 18. Output showing behavior of the Extra Stable mode of the tracking controller; Time versus Position (Radians); Time versus Position (Radians).

Drift Correction: Sample test results of the EKF filter for drift correction are shown in Figure 19. The graph is a result of simulating two consecutive harsh maneuvers on the Pitch rotational axes, with results of measured pitch angle, estimated pitch angle and their difference. This test is intended to validate EKF design and for the completion of the drift correction process by feeding back the calculated drift to the following stage of position controller. Figure 20 shows the simulation results for drone trajectory as a function of the entire flight controller: in the presence, and absence, of the drift correction mechanism. Figure 20a (left-most graph) represents the actual drone path (blue circles) versus the commanded path (red line) with drift correction active, while Figure 20b reveals drone path and commanded path for drift correction set to inactive.

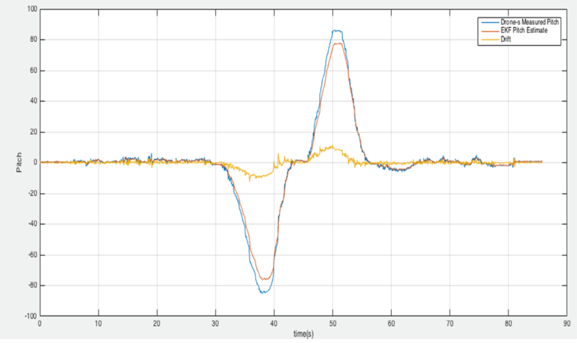

Figure 19. The behavior of the EKF for drift correction: these results show estimated pitch (in blue), measured pitch angle (in red) and the difference between these (in yellow).

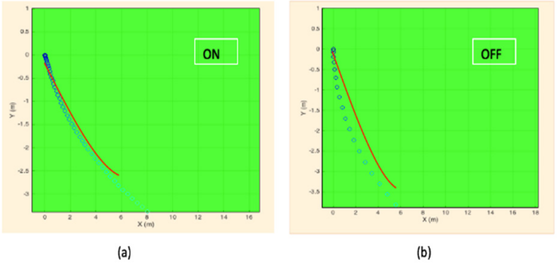

Figure 20. Simulation results of the output of the flight controller trajectory (planned path in red, actual path in blue) in the x-y plane with the presence of simulated wind disturbance, with (a) and without (b) drift correction.

FLC and System Results: Results of the FLC are generated in a simulation-based environment, utilizing functions within the Fuzzy Logic Toolbox embedded in MATLAB. Sample results representing the output model for variation in roll and pitch angle are shown in Figure 21. Sample results representing output configuration for variation in roll and pitch angle are visualized in Figure 22, and with variation in percentage of disturbance in Figure 23.

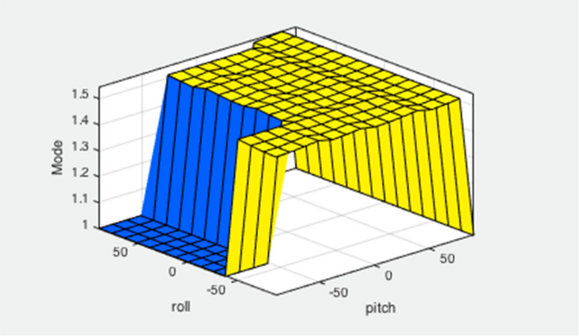

Figure 21. Results showing output "mode" as a function of variations in roll and pitch angles.

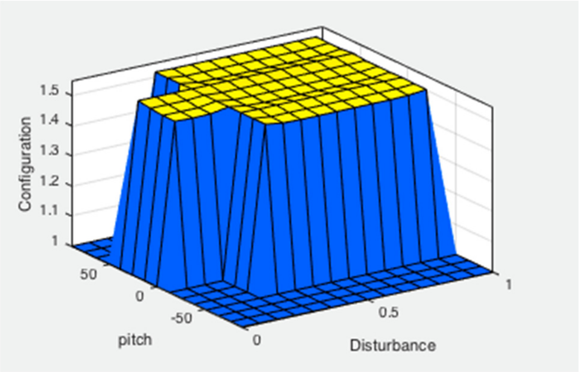

Figure 22. Results of output "configuration" as a function of variations in roll and pitch angles.

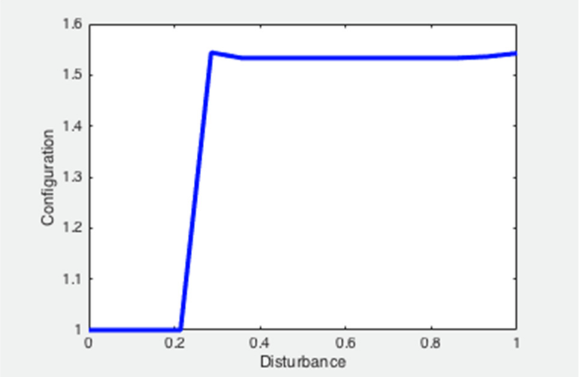

Figure 23. Results of output configuration based on the percentage of the disturbance.

4. Discussion

Position and Tracking Controller: The function of the Position Controller and Tracking Controller is to calculate and generate the correction control signals for the UAV. The Tracking Controller is designed to operate with two modes, wherein each mode should produce a correction signal in terms of the rotational angles, pitch, roll and yaw, as well as the altitude of the drone. As is evident in the results presented, this functionality is achieved. Both controllers exhibit fast, accurate and stable output for drone flight control correction, with the second mode exhibiting greater stability due to incorporation of a KF that increases its reliability in the presence of noise; appropriately minimizing the impact of this noise on flight control.

It is noted that Position Controller output is input to the tracking controller. Both controllers utilize the PID control technique and its customization. The controllers prove effective in determining the desired rotational rates when the input data are x, y, and y coordinate points. The output of the first stage is the desired correction in velocity of the body frame. The correction signal is adequately applied to the entire flight controller and enables successful control in the x, y, and z state coordinates of the drone. In the second stage of the controller, desired rotational rates are effectively input, along with the correction values of the first stage, in addition to measured angles of the drone. This resulted in appropriate, dynamic changing of the angle of the drone in accordance with the desired rates of change. Both stages of the Position Controller, in addition to the Tracking Controller, exhibited correct, responsive behavior for UAV flight control correction. The implementation of multiple flying modes of a robust controller widens the capabilities of a UAV by enabling it to fly and maneuver in challenging environments, as opposed to mono-mode flight controllers which require manual tuning for significant changes in flight capability and control technique.

Drift Correction: Drifts, caused by disturbances due to obstructions, are one of the major challenges faced by autonomous quadcopters during indoor flight. Non-linear lateral drifts are approximated in this work using a prediction model; EKF is applied in this capacity to provide an optimal solution for state estimation of this non-linear system of drifts. EKF is implemented for this stage of the flight controller for drift estimation and in response, to correct the UAV behavior after encountering such drifts. In this application, the EKF proved effective in estimating the pitch and roll angles of the UAV as isolated from the interference caused by the other controller in order to avoid the effect of additive noise. The state estimations of the EKF differed from the filtered measured angles and the difference between the two values was considered to be the estimated drift. As evident from the results, the implementation of the EKF provides a close estimate for the pitch and roll angles to those of the measured. This scenario was conducted in a simulated environment that did not introduce extreme surrounding environmental noise and as such did not cause significant drifts.

The expected behavior of the EKF is validated with the results obtained, as shown in Figures 19 and 20. Measured angles by the drone still contain process and control noise and as such have a larger magnitude than the values estimated by the EKF. However, this difference is sufficiently small. Overall, the drift correction mechanism reveals its capability of generating effective compensation for the lateral drifts: with an actual drone path suitably following the commanded (planned) trajectory (Figure 19b).

FLC: The flight controller has two flying modes and two kinematic configurations. The performance of each option of selected mode and configuration varies based on the input data and the flight environment. Due to this variability there is a need for another layer of control with the function to determine the flight mode and configuration of the UAV. The FLC was developed to provide this function. The FLC has three input signals of pitch, roll and disturbance, and two output values: mode and configuration. The FLC utilizes a set of 23 rule to determine the outputs based on the given inputs. Results show that by introducing the FLC greater flight stability is achieved by enabling selection of the most appropriate mode and configuration. The addition of the extra layer of control provides further self-governing capabilities, in which the FLC decides the suitable flight mode based on drone sensor readings and measured noise signals from the other controllers. The FLC controller successfully integrates the efforts of the other developed controllers, over existing methods [2], [14], [16].

5. Conclusions and Recommendations

A hybrid flight controller has been designed and tested in a simulation environment that achieves a high degree of stability and dexterity in maneuverability within an indoor location that is subject to non-linear drifts. For the Position Controller, a KF is applied within the mode that offers high in-flight stability, effectively rejecting induced noise and subsequently reducing the errors associated with Controller processing. Flight Control modes implemented achieve sound results in terms of stability within extreme disturbance conditions, in addition to sharp, fast, and accurate maneuvers under normal conditions. The Motor Control Mixer effectively maps the angle correction signals into a required throttle for each of the four motors, enabling flight in two separate configurations, dependent on desired throttle output. Non-linear drift estimation is effectively implemented using the EKF, in which compensation commands by the Position Controller are executed based on estimated drift calculation for pitch and roll angles. In addition to the PID and associated derivatives for motor control, the FLC proves useful in offering a further layer of flight control stability.

The proposed system can be further improved: by adding further Flight Controller modes for selection to enable greater specificity in flight behavior for different flight conditions. An automatic PID and PIDD^2 fine tuning mechanism may help to achieve even greater results in terms of system stability. Sensor fusion techniques, Bayesian Network and Central Limit Theorem, may be utilized in the IMU to achieve higher accuracy in angle measurements, which may lead to better state estimates of rotational angles and rates. KF and EKF design may be extended to include yaw and altitude UAV states, requiring addition of magnetometer and barometer sensors to attain these measurements. The FLC may be extended to process greater numbers of input, output and rule sets to provide more heightened control over the Flight Controller, particularly if the addition of Flight Controller modes is realized.

6. References

[1] M. Leichtfried, C. Kaltenriner, A. Mossel, H. Kaufmann. Autonomous Flight using a Smartphone as On-Board Processing Unit in GPS-Denied Environments, Proceedings of International Conference on Advances in Mobile Computing and Multimedia, pages 341-350, 2013.

[2] A. Sharma, A. Barve. Controlling of Quad-Rotor UAV using PID Controller and Fuzzy Logic Controller, International Journal of Electrical, Electronics and Computer Engineering, vol. 1, issue 2, pages 38-41.

[3] P.E. Pounds, D.R. Bersak, A.M. Dollar, Stability of Small-Scale UAV Helicopters and Quadrotors with Added Payload Mass under PID Control, Autonomous Robots, vol. 33, issue 1–2, pages 129-142, 2012.

[4] P. Salaskar, S. Paranjape, J. Reddy, A. Shah, Quadcopter–Obstacle Detection and Collision Avoidance, International Journal of Engineering Trends and Technology, vol. 17, issue 2, pages 84-87, 2014.

[5] S. Bouabdallah, Siegwart R, Full Control of a Quadrotor, *IEEE/RSJ International Conference on Intelligent Robots and Systems,* pages 153-158, 2007.

[6] A.L. Salih, M. Moghavvemi, H.A. Mohamed, K.S. Gaeid, Flight PID Controller Design for a UAV Quadrotor, Scientific Research and Essays, vol. 5, issue 23, pages 3660-3667, 2012.

[7] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys, Autonomous Obstacle Avoidance and Maneuvering on a Vision-Guided MAV using On-Board Processing, Robotics and Automation (ICRA), 2011 IEEE international conference on, pages 2472-2477, 2011.

[8] J.C.V. Junior, J.C. De Paula, G.V. Leandro, M.C. Bonfim, Stability Control of a Quad-Rotor using a PID Controller, Brazilian Journal of Instrumentation and Control, vol. 1, issue 1, pages 15-20, 2013.

[9] B. Kada, Y. Ghazzawi, Robust PID Controller Design for an UAV Flight Control System, Proceedings of the World Congress on Engineering and Computer Science, vol. 2, issue 1-6, 2011.

[10] L.C. Chien, Fuzzy Logic in Control Systems: Fuzzy Logic Controller- Part I, IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, issue 2, pages 404-418, 2011.

[11] I. Sa, P. Corke, Estimation and Control for an Open-Source Quadcopter, Proceedings of the Australasian Conference on Robotics and Automation, pages 7-9, 2011.

[12] M. Achtelik, T. Zhang, K. Kühnlenz, M. Buss, Visual Tracking and Control of a Quadcopter using a Stereo Camera System and Inertial Sensors, Mechatronics and Automation, IEEE, ICMA International Conference, pages 2863-2869, 2009.

[13] M. Hehn, R. D'Andrea, Real-Time Trajectory Generation for Quadcopters, Robotics, IEEE Transactions on, vol. 31, issue 4, pages 877-892, 2011.

[14] A.K. Gaurav, A. Kaur, Comparison Between Conventional PID and Fuzzy Logic Controller for Liquid Flow Control: Performance Evaluation of Fuzzy Logic and PID Controller by using MATLAB/Simulink, International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 1, issue 1, pages 84-88, 2012.

[15] Y. Al-Younes, M.A. Jarrah, Attitude Stabilization of Quadrotor UAV using Backstepping Fuzzy Logic and Backstepping Least-Mean-Square Controllers, Mechatronics and Its Applications, ISMA 5th International Symposium, pages 1–11, 2008.

[16] C.N. Hamdani, E.A.K. Rusdhianto, D.E. Iskandar. Perancangan Autonomous Landing Pada Quadcopter Menggunakan Behavior-Based Intelligent Fuzzy Control, Journal Teknik ITS, vol. 2, issue 2, pages E63-E68, 2013.

[17] L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys, Pixhawk: a System for Autonomous Flight using Onboard Computer Vision, Robotics and automation (ICRA), 2011 IEEE international conference on, pages 2992-2997, 2011.

Biographies

Dr. Catherine Todd received her B.Eng. (Electrical) and PhD. (Electrical) from the University of Wollongong, Australia. Currently she is working as an Assistant Professor of Engineering at Hawaii Pacific University. Her research interests include medical image processing, control systems and real-time simulation.

Dr. Stefano Fasciani is currently an Assistant Professor of Engineering at University of Wollongong in Dubai. He is doing research work in sound engineering. His main research interests are in voice processing, noise cancellation, HCI, acoustic and electronic engineering.

Mr. Hussien Koujan is working in Industry as an Electrical Engineer at Altorath International Engineering Consultants, having graduated from the University of Wollongong in Dubai with a Bachelor of Engineering (Electrical).